

AD-A053 546

MOORE SCHOOL OF ELECTRICAL ENGINEERING PHILADELPHIA PA
MOBILE COMMUNICATION NETWORKS WITH RANGE RESTRICTED CHANNELS. (U)
DEC 75 D A WALLEN

F/G 17/2.1

UNCLASSIFIED

NL

OF
AD A053546



END
DATE
FILMED

6-78

DDC

AD A 053546

University of Pennsylvania

THE MOORE SCHOOL OF ELECTRICAL ENGINEERING

MOBILE COMMUNICATION NETWORKS
WITH RANGE RESTRICTED CHANNELS.

David A. Wallen

55 p.

Master's thesis,

A thesis submitted to the Faculty of The Moore School of
Electrical Engineering in partial fulfillment of the require-
ments for the degree of Master of Science in Engineering
(for graduate work in Computer and Information Sciences).

Philadelphia, Pennsylvania

Dec 1975

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

DDC
RECEIVED
MAY 4 1978
RECEIVED
A

AD No. 1
DDC FILE COPY

237 000

self

University of Pennsylvania
THE MOORE SCHOOL OF ELECTRICAL ENGINEERING

Title of thesis: Mobile Communication Networks
With Range Restricted Channels

Abstract:

A graph model of a communication network is considered whose nodes represent the communication stations and whose links represent the communication channels. The links are undirected and there are no parallel links or self loops. The nodes are assumed to be 'mobile'; that is, their positions vary with time.

A procedure is presented which translates the network from a given initial position to a given terminal position and insures that at each intermediate position the network remains in one connected component subject to certain restrictions. Reliability functions are presented which allow the network elements, both links and nodes, to assume varying failure probabilities. At each network position, the survivability of the network is determined as a function of the link and node survivabilities by subjecting the network to a series of random attacks.

Degree and date of degree: Master of Science in Engineering
(for graduate work in Computer and Information Sciences).

December 1975


Signed:

David A. Hallen
AUTHOR

D. D. Behrouian
FACULTY SUPERVISOR

Table of Contents

| | <u>Page</u> |
|---|-------------|
| Index of Tables and Figures | i |
| Chapter 1 Related Results from the Literature | 1 |
| Chapter 2 Mobile Networks | 4 |
| 2.1 Introduction | 4 |
| 2.2 Problem statement | 4 |
| 2.3 Overview | 5 |
| 2.4 An application of mobile networks | 5 |
| Chapter 3 The Network Translation Algorithm | 8 |
| 3.1 The "Components Algorithm" | 8 |
| 3.2 Symbology | 8 |
| 3.3 The network translation algorithm | 10 |
| Chapter 4 Node Movement | 16 |
| 4.1 Introduction | 16 |
| 4.2 Choosing which nodes to move | 17 |
| 4.3 Generating a connected component | 18 |
| 4.4 Increasing the connectivity | 20 |
| 4.5 Maintaining a minimum separation distance | 21 |
| Chapter 5 Mobile Network Reliability | 23 |
| 5.1 Hardening network elements | 23 |
| 5.2 Node reliability | 24 |
| 5.3 Link reliability | 24 |
| Chapter 6 Results and Conclusions | 27 |
| 6.1 The mobile networks tested | 27 |
| 6.2 A standard for comparison of results | 27 |
| 6.3 Mobile network reliability results | 31 |
| 6.4 Conclusions | 34 |
| 6.5 Areas for future development | 35 |
| Bibliography | 38 |
| Appendix A Program Listing | A-1 |

| | |
|--|---|
| <i>Letter on file</i> | |
| <small>ADMINISTRATIVE/AVAILABILITY CODES</small> | |
| <small>AVAIL. AND/OR SPECIAL</small> | <small>OTHER</small> |
| A |  |

Index of Tables and Figures

| | <u>Page</u> |
|---|-------------|
| Figures: | |
| I. Network Translation Algorithm Flowchart | 13 |
| II. An Eight Node Mobile Network | 29 |
| III. A Twenty-five Node Mobile Network | 30 |
| IV. Network Survivability as a Function of Link Reliability | 32 |
| V. Network Survivability as a Function of Node Reliability | 33 |
| VI. Vulnerable Networks with Minimum Node Degree | 37 |
| Tables: | |
| I. Bounds on the Node Reliability | 25 |
| II. Bounds on the Link Reliability | 26 |

Chapter 1

Related Results from the Literature

Several authors have developed procedures for synthesizing highly reliable or invulnerable networks. The work of some of these authors is discussed briefly below in order to gain an understanding of the background of this paper.

Amin and Hakimi (AM1) prove that for a graph, G , of n nodes and e links, the network of G , denoted $N(G)$, is least vulnerable if the connectivity, w , is maximum over all graphs with n nodes and e links. They prove further that $N(G)$ is maximally survivable if the independence number, $\beta(G)$, is minimum, where the independence number is the cardinality of the maximum independent set, and a set $S \subseteq N$, the set of nodes, is an independent set of G if no two nodes in S are adjacent in G . Amin and Hakimi then consider the construction of G with a given connectivity and a minimum independence number in order to increase the network survivability.

Boesch and Thomas (B02 and B03) consider the connectivity and the cohesion, λ , and they prove that an optimal damage resistant network is one in which $w = \lambda = 2\frac{e}{n}$. Since any optimal damage resistant network must be regular, they construct such networks for given values of n and e by the union of edge disjoint Hamiltonian cycles. Boesch and Felzer (B01) expand this result to enumerate classes of optimal damage resistant networks.

Algorithms for the construction of optimal damage resistant networks are given by Chou and Frank (CH1 and FK5)

such that the number of branch disjoint paths between n_i and n_j is at least r_{ij} , where r_{ij} is an element of the branch redundancy matrix, $R=\{r_{ij}\}$. Using the result of Boesch and Thomas, Chou and Frank form Hamiltonian cycles of first order in their algorithms. They show that if all the node degrees, d_i for $i=1, \dots, n$, are even then the Hamiltonian cycles need not be restricted to first order.

Frank (FK1) generates a class of networks by a random process where all of the networks are symmetric in probability, develops a model for an attack strategy, and considers the survivability of the networks generated. Here, he considers the effects of uniform and nonuniform links, and links having a distance bias. In a later work, he (FK2) considers a class of networks having perfectly reliable links and nodes having a probability p of existing. The survivability of the network is examined for p close to unity.

In two more recent articles, Frank (FK3 and FK4) examines command and control communication systems. His network analysis begins with modeling the physical environment and ends with the solution of the analysis problems. Several measures of survivability are considered, including the number of nodes in the largest communicating section of the network after the attack.

Fratta, et al, (FR1) consider the design of networks in which the connection probability for the terminal node pairs in the worst conditions exceed some predetermined value. In their study the nodes of the network are completely reliable

while the links are nonuniformly unreliable.

Hänsler (HA1) presents an algorithm that recursively calculates the probability that all paths between two nodes in a communication network have been interrupted. His algorithm applies to networks with nonuniformly unreliable links and nodes. Hänsler, et al, (HA2) consider networks with not only unreliable links and nodes, but networks whose reliability also varies with the network topology. Their approach is to increase the reliability of the network by removing unreliable links and adding more reliable ones of lower cost. This approach follows that of Steiglitz, et al, (ST1) who also develop a procedure for designing optimally reliable networks subject to a cost constraint.

Wilkov (WK1) gives an approximation method for generating nonregular degree, maximally connected networks of minimum diameter, where the diameter is the maximum length of any shortest path in the graph. Williams (WL1) cites three ways to enhance the survivability of a network: hardening the elements, adjusting the location of the elements, and adding redundant elements. In his consideration the choice of methods is dictated by the costs involved, and he includes a procedure for approximating such costs.

The presentation above is by no means an exhaustive summary of the literature dealing with survivable networks. The reader is directed to Wilkov (WK1) for an excellent summation of a wide range of authors' works. Wilkov also includes an extensive bibliography.

Chapter 2

Mobile Networks

2.1: Introduction

In this paper a class of networks termed "mobile networks" is studied. A mobile network is defined as a network whose nodes have positions which vary with time. The mobile networks studied are characterized by the following assumptions: the links of the network are undirected; the network has no parallel links; the network has no self loops; each node of the network has a given initial and terminal point.

2.2: Problem statement

The problem dealt with here can be divided into two major parts. First, develop a procedure which moves the network from its given initial position to its given terminal position subject to the following restrictions:

- 1) The links have a fixed maximum length.
- 2) The nodes may move no more than a fixed maximum distance during one iteration.
- 3) Only a fixed maximum percentage of the nodes may move at one time.
- 4) The nodes may not assume intermediate positions coincident with other nodes.

The second part of the problem is to determine the survivability of the network at its initial and terminal position and at each intermediate position. For this analysis both the links and the nodes are assumed to be nonuniformly unreliable. The reliability of a node is inversely proportional

to its degree, and the reliability of a link is inversely proportional to its length. The network survivability will be determined by subjecting the network to a series of random attacks and computing the percentage of these attacks in which all of the surviving nodes remain in one connected component.

2.3: Overview

This paper presents a solution to the problem stated above. Chapter 3 describes the algorithm this author developed which successfully translates the mobile network from the initial to the terminal position while calculating the network survivability after each intermediate movement. Chapter 4 gives a detailed explanation of the portions of the algorithm which perform the actual movement of the nodes. The criteria used to determine which nodes to move is also explained. Chapter 5 describes the representative networks chosen to test the algorithm. For each such network, upper and lower bounds on the link and node reliability are calculated and shown in Chapter 5. Chapter 6 gives the results of the survivability analysis for the networks tested and the conclusions drawn from this study.

2.4: An application of mobile networks

When using a computer as an aid in solving such a network problem, it is understood that the original problem has been reduced to a graph model of the network. Indeed, the problem statement of Section 2.2 mentions only "links" and "nodes" and the attributes they are given in this particular problem. Thus the author has presented a graph model--but of what?

While there are many possible applications of mobile networks, only one will be presented here.

Assume that a number of military units are advancing toward their enemy. The enemy is believed to have no prior knowledge of the unit locations although he undoubtedly has a limited ability to detect these positions. During the advance, it is vitally important that the units be able to communicate with each other, either directly or by relaying messages.

This situation is perfectly suited for the mobile network model. Each unit, or station, in a communication network can be represented as a node in the graph. The communication channel joining two stations in the network can be represented by a link in the graph.

One restriction on the mobile networks imposed by the problem statement is that the links have a fixed maximum length. In Chapter 1 reference was made to Frank (FK1) who discusses networks whose links have a distance bias. Both of these concepts readily fit the military communication network application. The maximum length corresponds to the maximum transmission distance of the transmitter at each station. The distance bias corresponds to the lessening of reliability of the transmitted signal as its distance from the source increases. This lessening of reliability results from signal weakness, terrain interference, enemy jamming, etc. Relating these two concepts, the maximum transmission distance is defined as the range corresponding to the signal threshold

reliability. Beyond this range (or below this threshold) the communication quality via the channel is considered to be unacceptable.

The nodes of the network, rather than being perfectly reliable, have a survivability which is inversely proportional to their degree. For the communication network, the degree of a node represents the number of units within the maximum transmission range of a given unit. As the number of units within this range increases, the enemy's ability to detect the unit's location also increases--due not only to the increase in his target density but also to the probable increase in message traffic. Therefore, as the enemy's probability of success increases, the survivability of the unit decreases. Because of this increase in vulnerability of densely populated areas, the restriction that no two nodes in the network model may be coincident applies to the communication network. By extending the restriction, a minimum separation distance is established which helps position the nodes over a larger area, thus decreasing the density of the corresponding military units and increasing their survivability.

The network survivability criterion used here also applies to the military communication network. After an attack, the surviving units must be able to react to orders initiated by the commander. Autonomous units--those units which survive but cannot communicate with the commander--reduce his effectiveness since he is forced to assume that those units were also destroyed.

Chapter 3

The Network Translation Algorithm

3.1: The "Components Algorithm"

The algorithm presented in this paper incorporates an algorithm from the literature. Network connectivity is computed by determining the number of components in the network after it has been subjected to a random attack. This procedure is accomplished using Frank's "Components Algorithm" (FK4). Frank developed the procedure in order to determine the number of components remaining in a network when nodes and nodes are subject to failure after the network has been the target of a random attack. As described earlier, the mobile networks clearly correlate to Frank's network.

One modification of the "Components Algorithm" was made. After determining which nodes will be moved, temporary positions are calculated for these nodes and the network in this temporary position must be checked to insure that it is connected. This check is made rapidly using the "Components Algorithm" and bypassing the random attack portions of it. The bypass is accomplished through the use of a logical flag which, when Off (FALSE), directs the procedure to determine the number of components in a deterministic network and, when On (TRUE), directs the procedure to determine the number of components in a probabilistic network which has sustained a random attack.

3.2: Symbolology

A listing of the steps in the Network Translation

Algorithm appears in Section 3.3. A flowchart of the procedure is given in Figure I. The following symbol definitions apply to symbols found both in the algorithm and in the flowchart.

| | |
|-------------|--|
| n | the number of nodes in the network |
| D_n | the remaining distance each node must travel |
| F_N | terminal network position |
| d_F | minimum node degree in F_N |
| I_N | initial network position |
| d_I | minimum node degree in I_N |
| T_N | temporary network position |
| d_T | minimum node degree in T_N |
| d_N | minimum network degree |
| C_N | current network position |
| R_C | reliability of C_N |
| r | number of random attacks |
| s | number of random attacks C_N survives |
| nt | maximum number of nodes moving per iteration |
| M_{nt} | nodes that move in current iteration |
| MA_α | unmoved nodes not in main component |
| MB_β | moved nodes not in main component |
| V_{nv} | nodes not in main component after move |
| t | time |
| t_{max} | maximum allowable time |
| hf_n | multiplication factor |
| mf | maximum value of hf |
| D_{ij} | distance between nodes i and j |

rn maximum transmission range
md maximum movement distance
 σ minimum separation distance

3.3: The network translation algorithm

0. Preliminary calculations.

- a) Calculate D_i for $i=1, \dots, n$.
- b) If F_N is not connected, HALT. Calculate d_F .
- c) If I_N is not connected, HALT. Calculate d_I .
- d) $d_N = \min\{d_F, d_I\}$.
- e) Set $t = 0$ and $C_N = I_N$.

1. Random attack.

- a) Set $s = 0$. Conduct r random attacks on C_N . If after the i th attack the number of components in C_N is one, set $s = s + 1$.
- b) $R_C = \frac{s}{r}$.
- c) If $t > t_{\max}$, HALT.

2. Node movement.

- a) If $C_N = F_N$, HALT.
- b) $hf_i = 0$ for $i=1, \dots, n$.
- c) Let M_{nt} be the nodes with the largest D_n .
- d) Calculate T_N by moving M_{nt} a distance $2^{(-hf_i)} md$ closer to F_i for $i=1, \dots, nt$.
- e) Insure that no nodes in T_N are closer than σ .

3. Temporary position checkout.

- a) Is T_N connected? If so, set $ic = 0$ and go to 3b).
 - 1) Separate V_{nv} into MA_α and MB_β .
 - 2) Set $MO_{nt-nv} = M_{nt} - V_{nv}$. Sort d_i for $i \in MO$ in

descending order. Let $no = nt - nv$.

3) Do iteratively for $i=1, \dots, nv$. If $i \in MB$ go to b] below.

a] $i \in MA$:

1] $j = n(d_k)$ where $k = no - i$.

2] $T_j = C_j$.

3] $M_k = MA_i$.

4] $hf_i = 0$.

5] Move i $2^{(-hf_i)}$ md closer to F_i . Store the result in T_i .

6] Set $ic = 0$ and go to 3b).

b] $i \in MB$:

1] $hf_i = hf_i + 1$.

2] Move i closer to F_i by $2^{(-hf_i)}$ md. Store the result in T_i .

3] Set $ic = 0$.

b) If $d_T \geq d_N$ go to 4.

1) $ic = ic + 1$.

2) Let $j = nt - ic$, and set $T_j = C_j$.

3) Let i be such that $d_i < d_N$. If $i \in M_{nt}$, go to b].

a] Move i closer to F_i by $2^{(-hf_i)}$ md. Store the result in T_i , and go to 3b).

b] $hf_i = hf_i + 1$. If $hf_i \leq mh$, go to a] above.

1] Move i from C_i closer to F_i by $\frac{1}{2}$ md. Store the result in T_i .

2] Locate a node k such that if $D_{ik} \leq rn$ then $d_i \geq d_N$.

3] Move i toward k until $D_{ik} \leq rn$.

4] Go to 3b).

4. Preparation for the next time step.

a) $t = t + 1$.

b) Set $C_N = T_N$.

c) Go to 1.

Network Translation Algorithm Flowchart

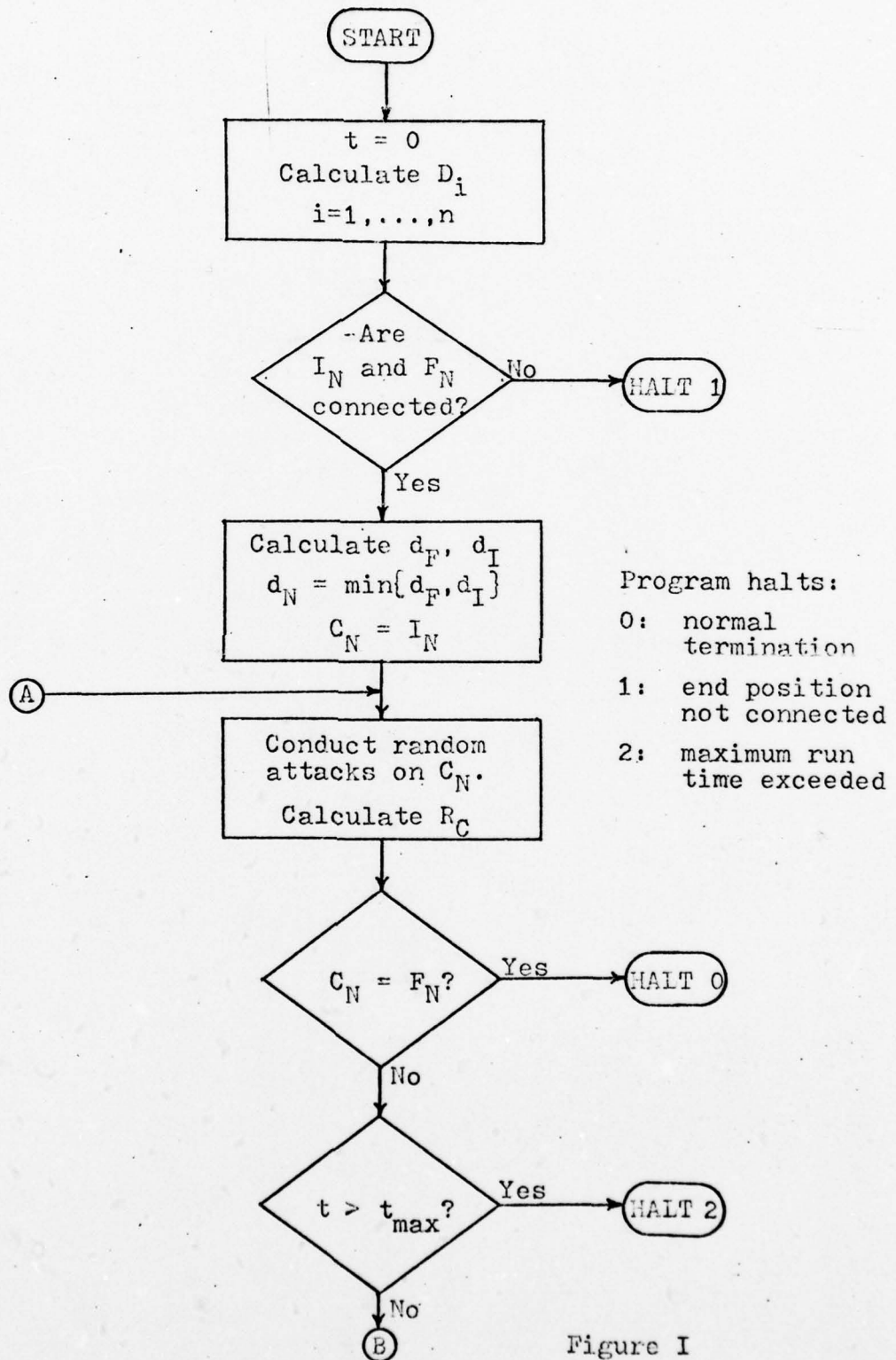


Figure I

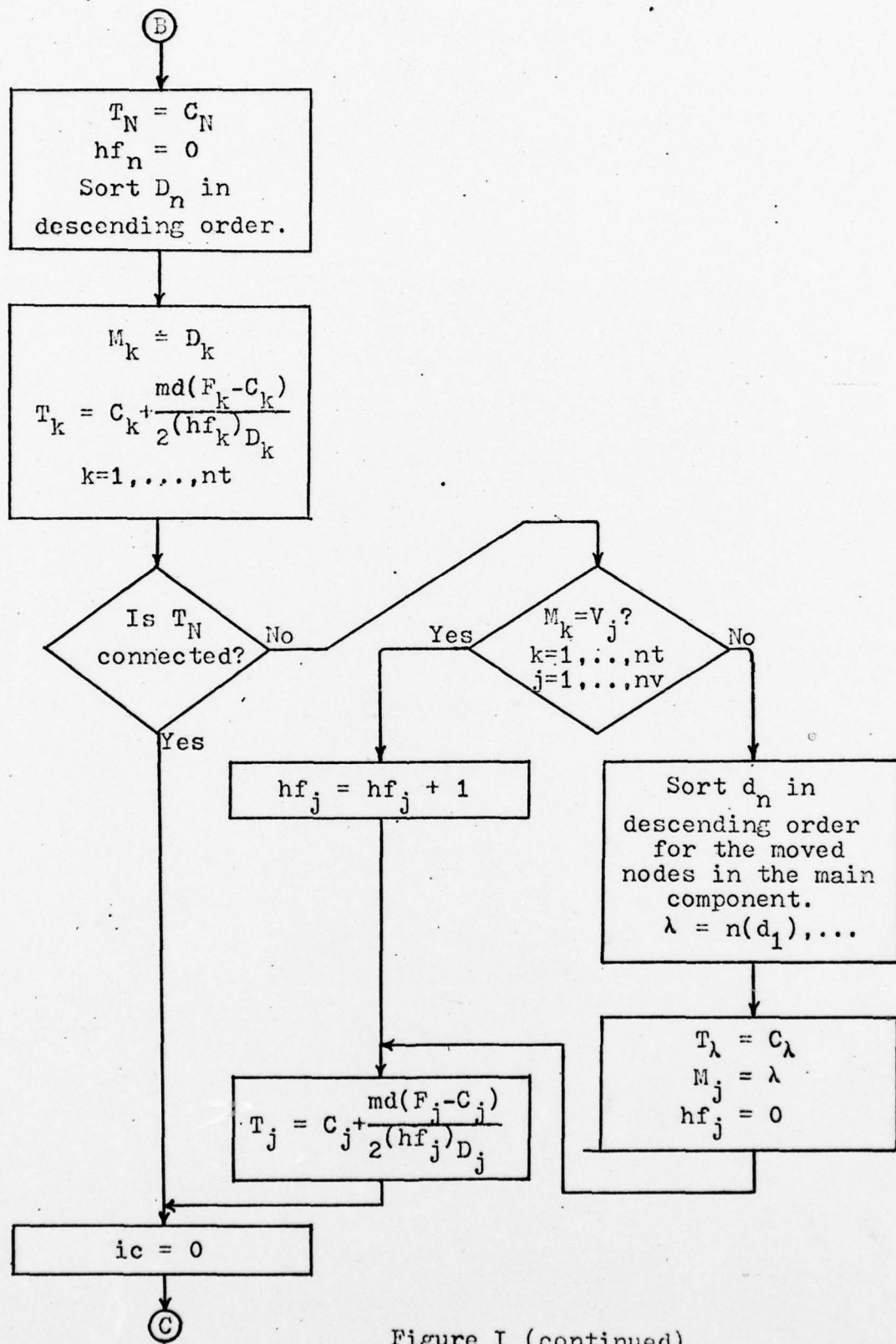


Figure I (continued)

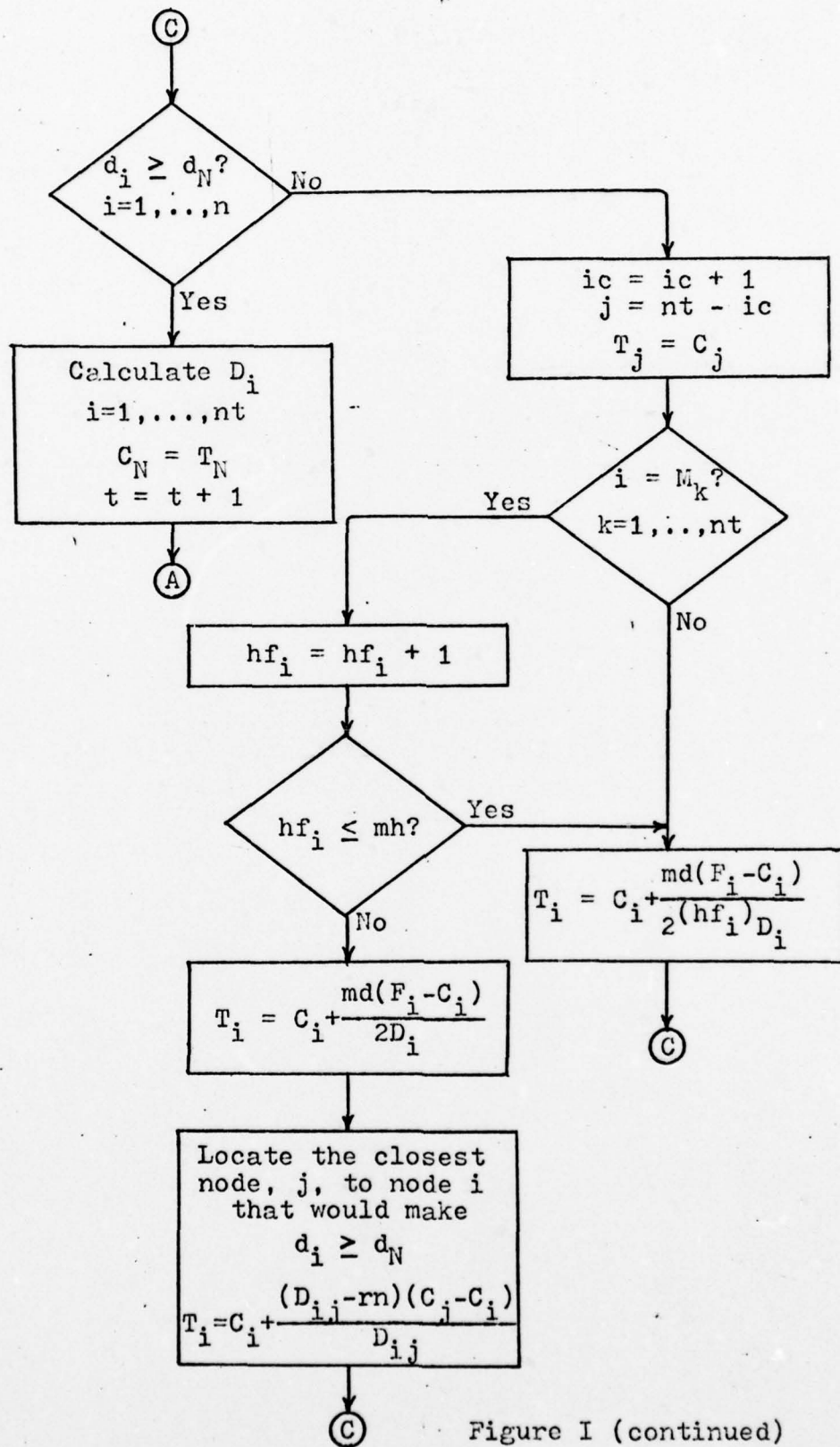


Figure I (continued)

Chapter 4

Node Movement

4.1: Introduction

The most important steps of the network translation algorithm--and the most difficult steps to comprehend--are the ones dealing with the movement of nodes and the checkout of the temporary positions generated. Reference was made in Chapter 1 to Williams' (W1) three methods for enhancing network survivability. Recall that the methods are:

- 1) Physically harden the network elements.
- 2) Adjust the location of the network elements.
- 3) Add redundant network elements.

The addition of redundant network elements is considered an impractical means of improving network survivability for the mobile networks studied here. This chapter will describe how the algorithm adjusts the location of the network elements to improve survivability, and Chapter 5 will deal with the problem of hardening the network elements.

Amin and Hakimi (AM1) proved that the survivability of a network can be increased by increasing the connectivity. The connectivity is increased by adding links to those nodes of minimum degree. Unfortunately, in mobile networks with range restricted links, one cannot simply add links to increase the connectivity since all of the links meeting the range restriction already appear in the network. This author did use the approach of Amin and Hakimi, however, in conjunction with Williams' proposal to adjust the location of the

network elements. As a result, the survivability of the mobile networks tested is increased by moving the nodes in such a way as to increase the network connectivity. Since the mobile networks studied have fixed initial and terminal positions, the connectivity of each of these end positions can be easily determined. These two connectivities are then compared, and their minimum value is defined to be the minimum network connectivity, or simply the network connectivity. Thus, at the intermediate positions--those positions where the location of the elements can be varied--each node is required to have a degree at least as great as the network connectivity.

4.2: Choosing which nodes to move

The network translation algorithm was formulated in such a way that it attempts to move the network from its initial position to its terminal position in the minimum number of iterations. In this context, an iteration is a movement of nt nodes where nt corresponds to the maximum percentage of nodes allowed to move concurrently. Since the most direct route from one point to another is simply a straight line, the algorithm will, when possible, adhere to straight line motion for each node moved. As will be discussed later, there are two possible situations which result in nonlinear motion.

The basic criterion for selecting which nodes to move is the distance between a node's current position and its terminal position. This distance is calculated by

$$D_{pq} = \left[\sum_{i=1}^d (x_{iq} - x_{ip})^2 \right]^{\frac{1}{2}}$$

where d is the number of dimensions and p and q are the points

between which the distance is being measured. The algorithm locates those n nodes with the greatest remaining distance, D , and translates them the maximum allowable movement distance, md , along the line connecting their current position and their terminal position.

The calculated positions of the nodes moved, along with the current positions of the nodes not moved, represent the temporary network position. Two tests are conducted on the network's temporary position: (a) is it in one connected component; and (b) is the degree of each node at least as great as the network connectivity. Sections 4.3 and 4.4 explain the procedures used when these tests fail.

4.3: Generating a connected component

In a mobile network where less than 100% of the nodes move in each iteration, it is quite possible that some motion will result in disconnecting the network. There are three cases that will dissolve the network into more than one component:

Case 1: A portion of the moving nodes move beyond the range of the main component.

Case 2: A portion of the nonmoving nodes is "stranded" as the main component moves away from it.

Case 3: Some combination of Case 1 and Case 2.

The discussion below will focus first on a Case 1 node and then on a Case 2 node. Discussing Case 3 would be redundant since it employs the techniques of both Case 1 and Case 2.

The distance a node moves during an iteration is given by $2^{(-hf)}md$, where md is the maximum allowable movement distance and hf is a multiplication factor. Initially, hf is set to zero for each node and the moving nodes are translated a distance 2^0md , or simply md . After the initial movement, if the network is no longer connected, hf is incremented by one for each Case 1 node and the temporary position of each such node is recomputed. Since the Case 1 node has moved beyond the main component, halving the distance it moves tends to bring it back into the range of the main component. The halving process continues a finite number of times--a number determined by the user based on the maximum movement distance and the maximum range of the links.

A Case 2 node presents an entirely different problem. The nodes that had been adjacent to the Case 2 node connecting it with the main component have moved beyond its link range. The problem would be simply resolved by moving the Case 2 node; however moving it will increase the number of nodes moved to a value greater than the maximum allowed. Therefore some node which did move must be moved back to the position it maintained at the start of the iteration before the Case 2 node can move. A criterion for selecting the node that must move back is needed. Consider the degree of the moved nodes: a node of high degree has a lower likelihood of being in a "critical" position--that is, a position which makes it the lone connector between some other node and the main component. Using this reasoning, the network translation

algorithm finds the node of highest degree among the moved nodes and returns it to the position it held at the beginning of the iteration. The Case 2 node is then moved forward and the maximum percentage of nodes allowed to move is not exceeded.

Each time there is more than one component in the network, the Case 1 nodes are repositioned to place them in the main component before the Case 2 nodes are considered. One node is brought back into, or moved forward into, the main component at a time, and after each such adjustment the network is reevaluated in order to prevent unnecessary changes since correcting one node may have the side effect of correcting others.

4.4: Increasing the connectivity

After the algorithm verifies that the network is composed of one connected component, the problem of connectivity of individual nodes is studied. If there is a node whose degree is less than the network connectivity, the position of this node will be altered to increase its degree.

The node whose connectivity must be increased will move from its temporary position back toward the position it maintained at the start of the iteration, moving $\frac{1}{2}$ unit each step. After each such move, the degree of the node will be recalculated and the procedure will end if the degree is at least as great as the network connectivity. If the degree requirement is still not satisfied, the node continues moving backward in $\frac{1}{2}$ unit steps. The process stops if the node reaches a point

only $\frac{1}{2}$ unit from where it began the iteration. Having exhausted the possibility of each half unit interval within the maximum movement range along the line connecting its start point and its terminal point, the node enters a routine which results in nonlinear motion in order to satisfy the degree requirement on the node.

The node is moved from its starting point for the iteration a distance $\frac{1}{2}$ nd toward its terminal position. From this position it searches for the closest node that would raise its degree to at least the minimum network connectivity. When such a node is located, the moving node is translated along the line connecting its current position and the node found until the two nodes are separated by a distance less than or equal to the maximum link range.

The discussion above assumes that the node whose degree must be increased is one of the nodes moved during the iteration. If in fact it was not moved, it is treated as a Case 2 node, in the terminology defined in Section 4.3, and some other node is returned to its starting point so the Case 2 node can be moved to increase its degree.

4.5: Maintaining a minimum separation distance

Each time a node is moved, its distance from its neighbors is checked. If the distance between the moved node and any other node is less than the minimum allowable separation, the position of the moved node must be varied. The variation in movement resulting from this situation is the second instance of nonlinear motion contained within the network

translation algorithm.

The moved node is translated away from its two closest neighbor along the line connecting it with its neighbor until the distance between them is at least as great as the minimum separation distance. Since it is possible that such a shift in position will result in the total displacement of the moved node exceeding the maximum allowable movement distance, the distance moved is rechecked. If the maximum displacement has been exceeded, the node will move back toward its starting point until both the maximum movement restriction and the minimum separation restriction are satisfied.

Chapter 5

Mobile Network Reliability

5.1: Hardening network elements

One method cited by Williams (WL1) for increasing the network survivability is hardening the network elements. There are two basic ways to simulate the hardening of network elements for the purpose of network analysis. The more subtle approach is given by Frank (FK1). In his network, station s is destroyed if it is hit by K_s weapons. Similarly, channel c is destroyed if it is hit by K_c weapons. Frank assumes that each weapon can produce at most one hit, and the probability of a network element being hit by any given weapon is independent of the number of hits the element has already sustained. This approach relates the hardness of an element to the number of weapon hits required to totally destroy it. The effect of varying degrees of hardness can be compared by changing the values of K_s and K_c . Such an approach would be quite realistic for a military application since in a conventional warfare situation a unit would probably not be totally destroyed by a single weapon hit.

The second way to simulate hardening the network elements is the Monte Carlo method used by Baran (BA1). Baran tests the survivability of networks after establishing values for the node and link reliabilities. By varying these reliabilities various degrees of element hardness can be compared.

Despite the appropriateness of Frank's method to the application developed in Chapter 2, Baran's approach is used in

this paper. Frank's method is viewed as being too cumbersome for the analysis included here.

5.2: Node reliability

In the network translation algorithm, the survival probability of node i is given by

$$p_i = 1.0 - \left[\frac{(100.0 - t_n)10^{-2}}{n - 1} \right] d_i$$

where t_n is the node threshold reliability and d_i is the degree of node i .

Upper and lower bounds on the node reliability can be established. Since the reliability of the node is inversely proportional to its degree, the lower bound on the node reliability corresponds to a node of maximum degree. In the mobile networks considered here, where there are no parallel links and no self loops, the maximum degree of any node in a network of n nodes is $n-1$. The given value of the node threshold reliability is made to correspond to a degree of $n-1$. An upper bound on the node reliability corresponds to a node of minimum degree. As discussed earlier, the minimum degree of any node is the calculated value of the network connectivity.

The algorithm was tested on the two networks that will be presented in Chapter 6. The smaller network has 8 nodes and the larger network has 25 nodes. In each network, the minimum network connectivity is three. Table I gives upper limit values for the node survival probability for each network where the lower limit is the threshold reliability.

5.3: Link reliability

In the network translation algorithm, the survival pro-

Bounds on the Node Reliability

| <u>Lower</u> | <u>Upper</u> | |
|--------------|--------------|--------|
| | n = 8 | n = 25 |
| 70.0% | 87.14% | 96.25% |
| 80.0 | 91.43 | 97.50 |
| 90.0 | 95.71 | 98.75 |
| 100.0 | 100.00 | 100.00 |

Table I

bability of link j is given by

$$p_j = 1.0 - \left[\frac{(100.0 - t_1)10^{-2}}{rn} \right]^{D_{ik}}$$

where t_1 is the link threshold reliability, rn is the maximum link range, and D_{ik} is the distance between nodes i and k, the endpoints of the jth link.

Upper and lower bounds on the link reliability can also be established. Since the reliability of the link is inversely proportional to its length, the lower bound on link reliability corresponds to a link of maximum length. The upper bound on the link reliability corresponds to a link of minimum length where the minimum length is given by the minimum separation distance. For the two networks tested the maximum range was the same, as was the minimum separation. Table II gives upper limit values for the link survival probability for varying lower limits, where the lower limit is equal to the given value of the threshold reliability.

Bounds on the Link Reliability

| <u>Lower</u> | <u>Upper</u> |
|--------------|--------------|
| 50.0% | 93.75% |
| 60.0 | 95.00 |
| 70.0 | 96.25 |
| 80.0 | 97.50 |
| 90.0 | 98.75 |
| 100.0 | 100.00 |

Table II

Chapter 6

Results and Conclusions

6.1: The mobile networks tested

Two mobile networks were selected at random to check the network translation algorithm. Figure II shows the graph model of the initial and terminal positions of the mobile network of 8 nodes. The algorithm translated the network from its initial to its terminal position in five iterations which is the minimum number possible for the given parameters.

Figure III shows the graph model of the initial and terminal positions for the mobile network of 25 nodes. The algorithm completed the translation of the network in 13 iterations. The minimum number of iterations for the network under the given parameters is 12; however, only two nodes remained to be translated after 12 iterations and the greater remaining distance of these two nodes was 0.698 units.

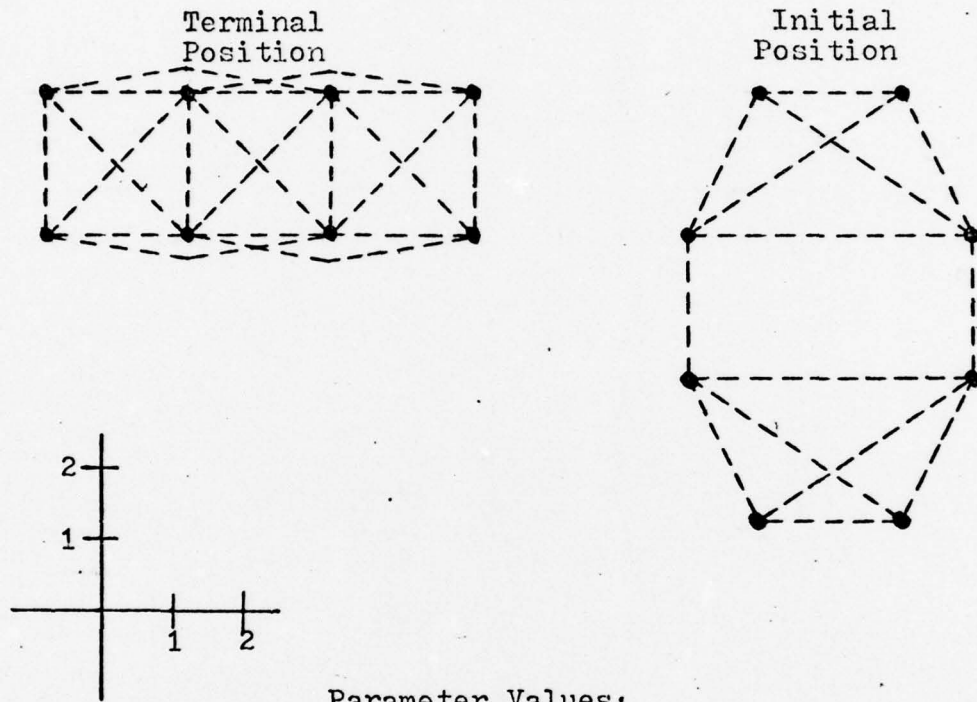
6.2: A standard for comparison of results

The network survivability criterion used to measure the survivability of mobile networks is similar to that used by Baran (BA1) in his analysis of distributed communication networks. He judged the survivability of a network by the percentage of nodes in connection with the largest single group of surviving nodes, and he considered small groups of isolated nodes to be ineffective. Baran showed, using Monte Carlo simulation, that a highly reliable network can be constructed using unreliable links when a satisfactory redundancy level is chosen. He defines the redundancy level as the link-to-node

ratio in an infinite size array. Baran proved that for perfectly reliable nodes, links of only 50% reliability cause little system degradation for redundancy three or four.

Because of the similarities between the mobile networks tested and Baran's distributed networks, it is possible to use his results to measure the accuracy of the results generated by the network translation algorithm. However, before proceeding an understanding of the differences in the two approaches must be gained. From Baran's definition of the redundancy level it is clear that the redundancy level of a node is twice the node's degree. For networks with fixed initial and terminal positions and range restricted links the connectivity can be determined but not increased. Only the connectivity at intermediate positions can be varied, and it cannot be reasonably maintained at a higher level than the minimum connectivity of the two end positions. As discussed earlier this procedure is used to establish the network connectivity. If the network connectivity is six or eight--corresponding to a redundancy level of three or four--then the result Baran proved concerning systems with links of 50% reliability could be checked. However, for a network connectivity of less than six, a higher degree of degradation must be expected. The networks evaluated by the network translation algorithm both have a network connectivity of three. When the results of the survivability tests on these networks are shown in Section 6.3 they will be compared to Baran's equivalent network with a redundancy level of 1.5.

An Eight Node Mobile Network



Parameter Values:

Number of nodes moving per iteration: 4

Maximum link range: 4.0

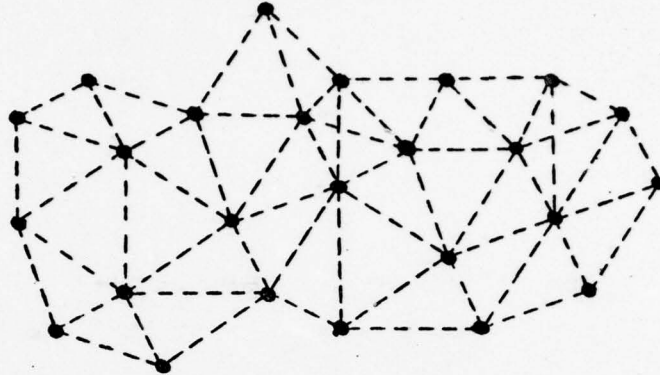
Maximum movement distance: 4.0

Minimum network connectivity: 3

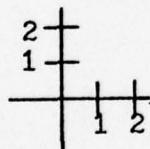
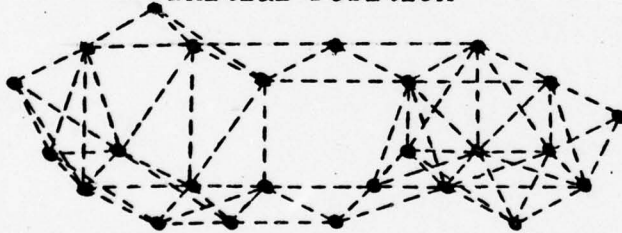
Figure II

A Twenty-five Node Mobile Network

Terminal Position



Initial Position



Parameter Values:

Number of nodes moving per iteration: 12
Maximum link range: 4.0
Maximum movement distance: 4.0
Minimum network connectivity: 3

Figure III

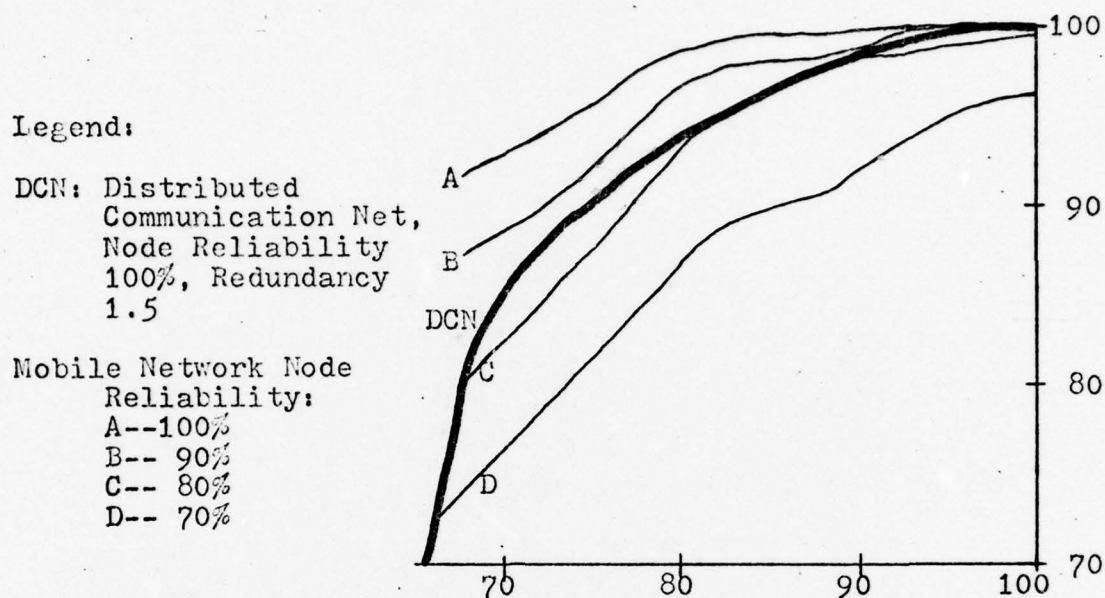
The way in which the link and node reliabilities are determined in the network translation algorithm will also cause the results generated to deviate from Baran's. If the link threshold reliability is 50%, only those links of maximum length will be 50% reliable. Clearly, not all of the links will be of maximum length. The links of less than maximum length will be more than 50% reliable, thus tending to reduce the network degradation. Similarly for any value of the node threshold reliability, only those nodes of maximum degree will exhibit the threshold reliability. Nodes of lower degree will, therefore, also reduce the network degradation.

6.3: Mobile network reliability results

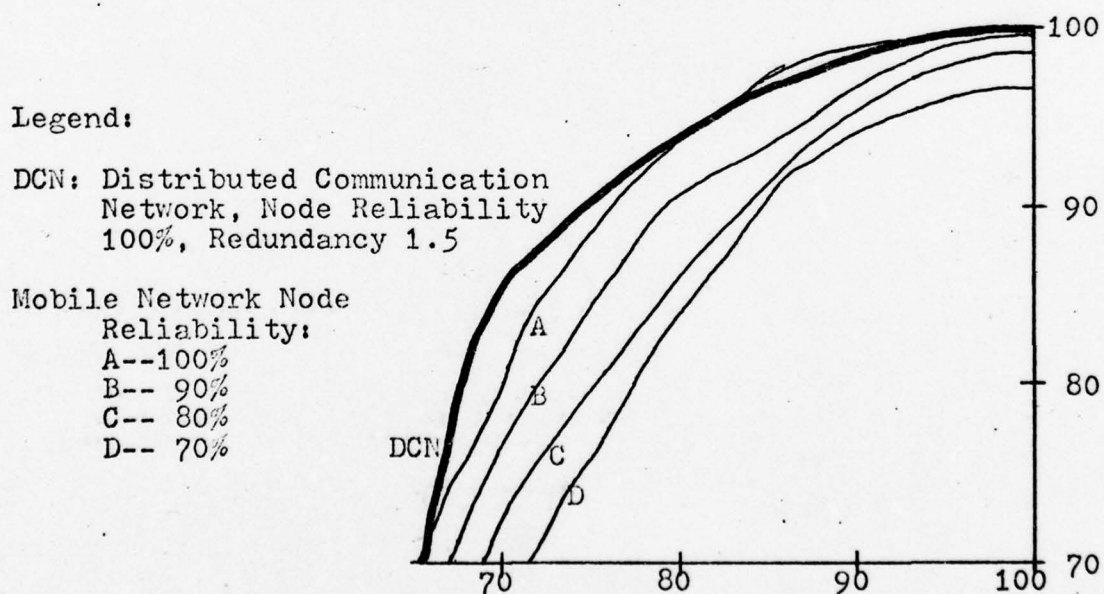
Figures IV and V are graphs of the mobile network survivability generated for varying values of link and node reliability, respectively. The results are shown for both the 8 and the 25 node networks as continuous functions. Clearly, these functional representations are broad extensions of a relatively small number of discrete samples. The continuous curves serve as a visual aid but have not been rigorously determined. In each figure, the curve labeled "DCN" corresponds to Baran's results for his Distributed Communication Networks.

The mobile network results shown in Figures IV and V are averages for each network. For given values of the link and node reliabilities, each network at each iteration underwent a series of 50 random attacks. The plotted curves correspond to the average calculated value of the network survivability over all iterations.

Network Survivability As a Function of Link Reliability



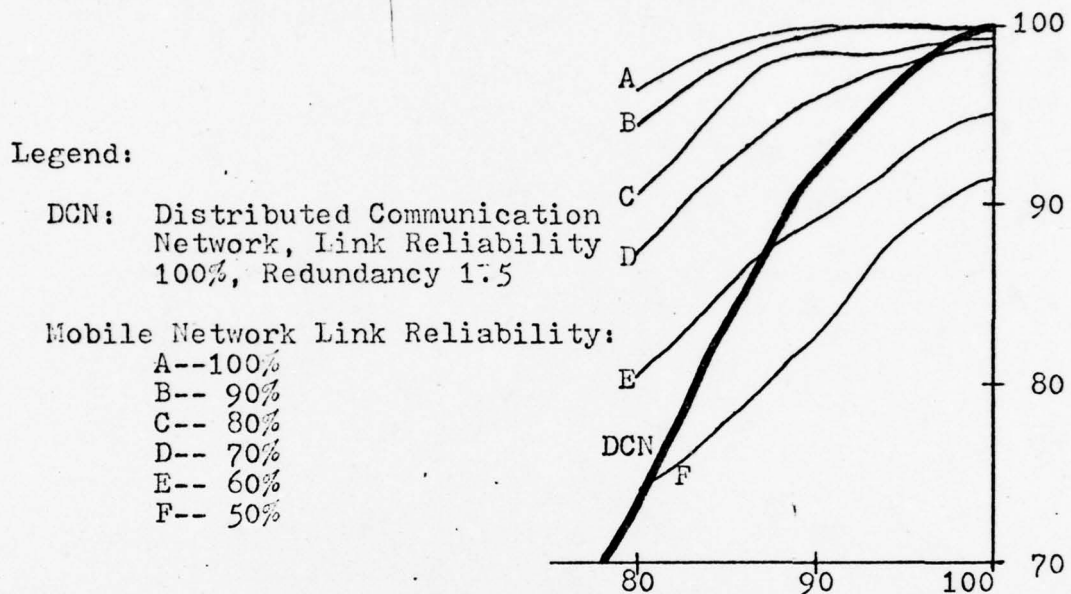
(a) Eight Node Network



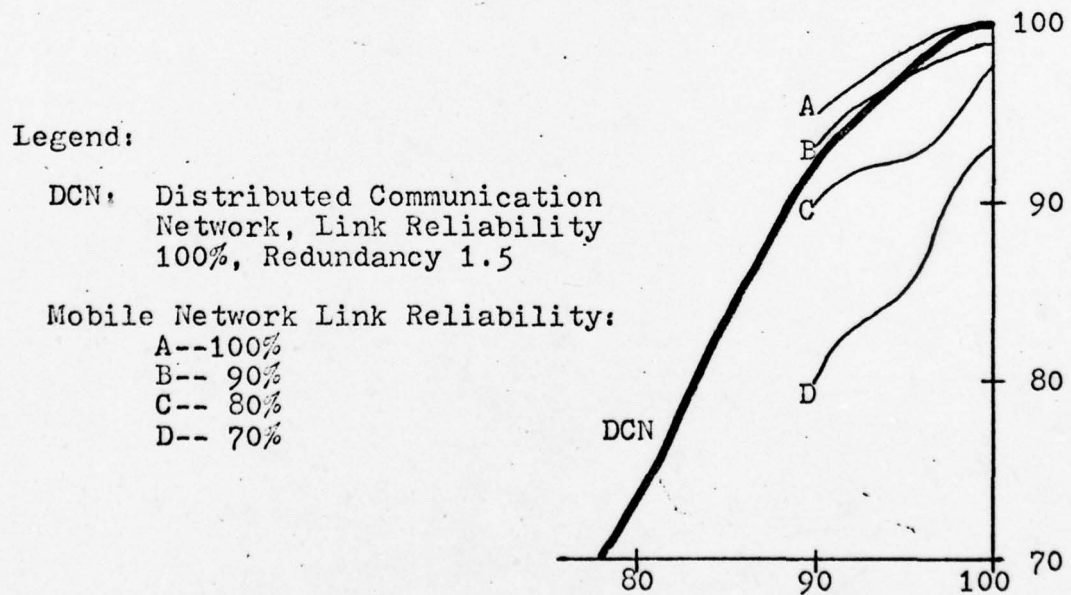
(b) Twenty-five Node Network

Figure IV

Network Survivability As a Function of Node Reliability



(a) Eight Node Network



(b) Twenty-five Node Network

Figure V

6.4: Conclusions

This paper presented an algorithm, the network translation algorithm, which successfully translates a mobile network from a fixed initial position to a fixed terminal position while maintaining an established minimum network connectivity at each intermediate iteration. Two representative sample networks were chosen to test the algorithm. Based on the results generated by these tests, the algorithm is felt to be very efficient in so far as the procedure progresses to a conclusion in an optimal number of iterations.

The reliability of the mobile networks closely parallels the reliability of the distributed networks of equivalent redundancy studied by Baran (BA1), as evidenced by the survivability curves in Figures IV and V. This similarity was expected since the requirement of a minimum degree on each node is the foundation of the definition of distributed networks.

It is also evident from Figure V that, for a small mobile network, a decrease in the node survivability does not affect the network survivability as much as does a corresponding reduction of the node reliability in a distributed network. The decrease in the slope of the network survivability curve for sufficiently small mobile networks is due to the fact that not all nodes are of equal degree as in distributed networks. As a result, the loss of any given node will not have as high a probability of disrupting the network.

It can also be seen by comparing the mobile network survivability curves of Figures IV and V that for corresponding values of the respective threshold reliabilities in

the eight node network, the nodes and links affect the network survivability in much the same way. This leads the author to hypothesize that if the main consideration in a sufficiently small mobile network is the cost of the elements, then the survivability of the network will not be greatly affected by choosing the lower cost element. For example, suppose that the cost of a node in the network is much greater than the cost of increasing the range of the links, where changing the restriction on the link range is the only feasible way to add links. Then the hypothesis is that it would be possible to reduce the number of nodes and find an increased range for the links that would realize a reduction in the network cost without adversely affecting the network survivability.

6.5: Areas for future development

Two aspects of mobile networks are viewed as areas in which further work might be useful. The first of these would involve the development of the hypothesis above to determine the effect on the network survivability induced by changing the number of nodes and the range of the links. These tests would also help determine the maximum size of the mobile network for which the survivability of the network would not be reduced when reducing the network cost.

The second area of possible study involves the movement of the nodes in the network translation algorithm. The algorithm positions the nodes such that the degree of each node is at least as great as the network connectivity. A different

criteria, which would perhaps increase the network survivability, would be to maximize--or at least set a lower bound on--the smallest cut-set at each iteration. The cut-set to be bounded could be a node cut-set, a link cut-set, or a combined link and node cut-set. Limiting the minimum node degree, which is the connectivity criterion used here, does not guarantee that the network will not assume the characteristics of a star graph at any iteration as shown in Figure VI(a). In a star graph the minimum node cut-set is one; therefore the network is highly vulnerable because the destruction of one node may destroy the network. Another highly vulnerable network which satisfies the minimum node degree criterion is shown in Figure VI(b) where the minimum link cut-set of one indicates that the network will be destroyed by destroying one critical link. Changing the connectivity criteria in the network translation algorithm could, therefore, increase the total network survivability by eliminating the possibility that at intermediate iterations the network will assume the characteristics exhibited in Figure VI.

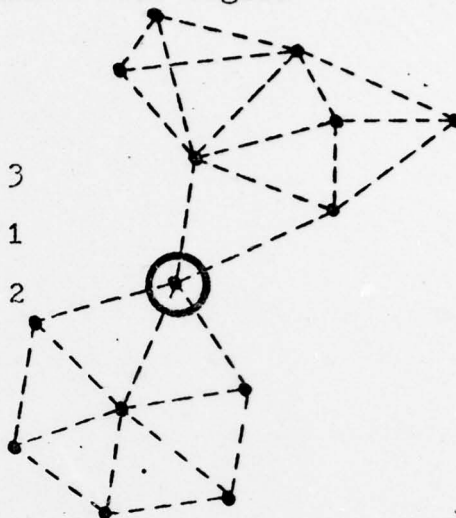
Vulnerable Networks

With Minimum Node Degree

Minimum node degree: 3

Minimum node cut-set: 1

Minimum link cut-set: 2



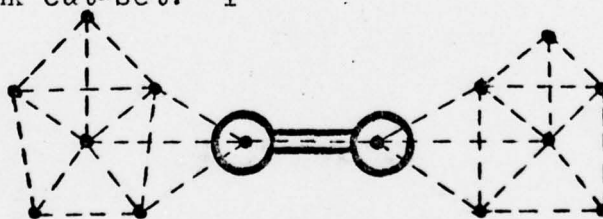
Critical node indicated

(a)

Minimum node degree: 3

Minimum node cut-set: 1

Minimum link cut-set: 1



Critical elements indicated

(b)

Figure VI

Bibliography

- AM1 A. T. Amin and S. L. Hakimi, "Graphs with Given Connectivity and Independence Number or Networks with Given Measures of Survivability", IEEE Transactions on Circuit Theory, vol. CT-20, pp. 2-10 (1973).
- BA1 P. Baran, "On Distributed Communications Networks", IEEE Transactions on Communication Systems, vol. CS-12, pp. 1-9 (1964).
- B01 F. T. Boesch and A. P. Felzer, "On the Minimum m-Degree Vulnerability Criterion", IEEE Transactions on Circuit Theory, vol. CT-18, pp. 224-228 (1971).
- B02 F. T. Boesch and R. E. Thomas, "Optimal Damage Resistant Communication Nets", IEEE International Communication Conference, vol. 4, pp. 688-693 (1968).
- B03 F. T. Boesch and R. E. Thomas, "On Graphs of Invulnerable Communication Nets", IEEE Transactions on Circuit Theory, vol. CT-17, pp. 183-192 (1970).
- CH1 W. Chou and H. Frank, "Survivable Communication Networks and the Terminal Capacity Matrix", IEEE Transactions on Circuit Theory, vol. CT-17, pp. 192-197 (1970).
- FK1 H. Frank, "Vulnerability of Communication Networks", IEEE Transactions on Communication Technology, vol. COM-15, pp. 778-789 (1967).
- FK2 H. Frank, "Maximally Reliable Node Weighted Graphs", Proceedings of the 3rd Annual Princeton Conference on Information Science and Systems, vol. 3, pp. 1-9 (1969).
- FK3 H. Frank, "Survivability Analysis of Command and Control Communication Networks--Part I", IEEE Transactions on Communication, vol. COM-22, pp. 589-595 (1974).
- FK4 H. Frank, "Survivability Analysis of Command and Control Communication Networks--Part II", IEEE Transactions on Communication, vol. COM-22, pp. 596-605 (1974).
- FK5 H. Frank and W. Chou, "Connectivity Considerations in the Design of Survivable Networks", IEEE Transactions on Circuit Theory, vol. CT-17, pp. 486-490 (1970).
- FR1 L. Fratta, F. Maffioli, and G. Tartara, "Synthesis of Communication Networks with Reliability of Links Exponentially Decreasing with their Length", IEEE International Communication Conference, vol. 7, pp. 39-33 - 39-37 (1971).

- HA1 E. Hänsler, "A Fast Recursive Algorithm to Calculate Reliability of a Communication Network", IEEE Transactions on Communication, vol. COM-20, pp. 637-640 (1972).
- HA2 E. Hänsler, G. K. McAuliffe, and R. S. Wilkov, "Optimizing the Reliability in Centralized Computer Networks", IEEE Transactions on Communication, vol. COM-20, pp. 640-644 (1972).
- PA1 S. C. Parikh and I. T. Frisch, "Finding the Most Reliable Routes in Communication Systems", IEEE Transactions on Communication Systems, vol. CS-11, pp. 402-406 (1963).
- ST1 K. Steiglitz, P. Weiner, and D. J. Kleitman, "The Design of Minimum-Cost Survivable Networks", IEEE Transactions on Circuit Theory, vol. CT-16, pp. 455-460 (1969).
- WK1 R. S. Wilkov, "Analysis and Design of Reliable Computer Networks", IEEE Transactions on Communication, vol. COM-20, pp. 660-678 (1972).
- WL1 T. G. Williams, "The Design of Survivable Communication Networks", IEEE Transactions on Communication Systems, vol. COM-11, pp. 230-241 (1963).

Appendix A
Program Listing

Contained in this appendix is a listing for the program written by this author to implement the network translation algorithm. The program was written in FORTRAN IV with WATFOR/WATFIV.

```

1      DIMENSION PSTR(50,4),PSTP(50,4),DISP(50),LINK(2,200),
2      DIST(50,50),PSLI(200),PSNO(50),NDEG(50),NRAN(50),
3      LRAN(200),RANN(50),RANL(200),PTM(50,4),NMOV(50),
4      AR(50,2),NVAR(50),ISTP(50),NMB(50),NOK(50),NAD(50)
2      REAL NFLG(50)/50*' '/' ,ASTAR/' '*' /,BLANK/' '/' ,MD
3      LOGICAL RANDOM,FLAG,FINAL,FIRST
4      DATA NS,LS/50,200/
5      DATA MD,MINP,NCASE,PC,RN,SEP/4.0,2,20,50.0,4.0,0.5/
6      READ (5,402) PCLN,PCND,NRAT
7      READ (5,403) N,ND,((PSTR(I,J),J=1,ND),I=1,N),
8      2 ((PSTP(I,J),J=1,ND),I=1,N)
      ICASE = 0

      C
      C
      C      CALCULATE THE TOTAL DISPLACEMENT OF EACH NODE.
9      DO 10 I = 1,N
10     NMOV(I) = 0
11     10  DISP(I) = DISTAN (NS,I,I,ND,PSTR,PSTP)
12     NT = INT(N*PC*1.E-2)
13     WRITE (6,501)
14     WRITE (6,502) (I,(PSTR(I,J),J=1,ND),(PSTP(I,J),
15     2 J=1,ND),I=1,N)
16     WRITE (6,515) PCND,PCLN,RN,N,NT,MD,NRAT
17     FCTL = (100.0 - PCLN)*1.E-2/RN
      FCTN = (100.0 - PCND)*1.E-2/(N-1.)

      C
      C
      C      INSURE THAT THE TERMINAL AND INITIAL POSITIONS ARE
      C      CONNECTED.
18     CALL ADJA (NS,LS,N,ND,RN,PSTP,DIST,LINK,NDEG,M)
19     MNDG = NDEG(1)
20     DO 20 I = 2,N
21     20  IF (NDEG(I).LT.MNDG) MNDG = NDEG(I)
22     RANDOM = .FALSE.
23     CALL COMON (NS,N,M,RANDOM,PSNO,PSLI,RANN,RANL,
24     2 LINK,LCOM,NVAR,NV)
25     IF (LCOM.NE.1) GO TO 150
26     CALL ADJA (NS,LS,N,ND,RN,PSTR,DIST,LINK,NDEG,M)
27     DO 30 I = 1,N
28     30  IF (NDEG(I).LT.MNDG) MNDG = NDEG(I)
29     WRITE (6,517) MNDG
30     CALL COMON (NS,N,M,RANDOM,PSNO,PSLI,RANN,RANL,
31     2 LINK,LCOM,NVAR,NV)
32     IF (LCOM.NE.1) GO TO 160
33     FINAL = .FALSE.

      C
      C
      C      BEGIN THE MAIN CALCULATIONS. CONDUCT A RANDOM
      C      ATTACK ON THE CURRENT NETWORK POSITION.
34     MO = NT
35     45  WRITE (6,503)
36     WRITE (6,504) (I,NFLG(I),(PSTR(I,J),J=1,ND),NDEG(I),
37     2 DISP(I),I=1,N)
38     WRITE (6,516)

```

```

36      DO 47 I = 1,N
37 47    NFLG(I) = BLANK
38      RANDOM = .TRUE.
39      DVD = 0.
40      TLR = 0.
41      TNR = 0.
42      DO 48 K = 1,M
43      I = LINK(1,K)
44      J = LINK(2,K)
45      PSLI(K) = 1. - FCTL*DIST(I,J)
46 48    TLR = TLR + PSLI(K)
47      AVLI = TLR/M
48      DO 49 I = 1,N
49      PSNO(I) = 1. - FCTN*NDEG(I)
50 49    TNR = TNR + PSNO(I)
51      AVNO = TNR/N
52      WRITE (6,521) AVNO,AVLI
53      DO 70 ITER = 1,NRAT
54      READ (5,401) (NRAN(I),I=1,N)
55      READ (5,401) (LRAN(LN),LN=1,M)
56      DO 50 LN = 1,M
57 50    RANL(LN) = LRAN(LN)*1.E-2
58      DO 60 I = 1,N
59 60    RANN(I) = NRAN(I)*1.E-2
60      CALL COMCON (NS,N,M,RANDOM,PSNO,PSLI,RANN,RANL,
2 LINK,LCOM,NVAR,NV)
61 70    IF (LCOM.EQ.1) DVD = DVD + 1.
62      REL = DVD/NRAT
63      WRITE (6,507) REL
64      IF (FINAL) GO TO 140
65      WRITE (6,508) ICASE
66      ICASE = ICASE + 1
67      IF (ICASE.GT.NCASE) GO TO 170
68      DO 69 I = 1,N
69 69    ISTD(I) = 0
70      RANDOM = .FALSE.

C
C
C
71      DO 71 I = 1,N
72      AR(I,1) = I
73      AR(I,2) = DISP(I)
74      DO 71 J = 1,ND
75 71    PTEM(I,J) = PSTR(I,J)
76      CALL SORT (NS,N,AR)
77      FIRST = .TRUE.
78      DO 77 J = 1,MO
79      NMOV(J) = AR(J,1)
80      I = NMOV(J)
81 77    CALL TEMIN (NS,N,ND,MD,I,DISP(I),ISTD(I),SEP,PSTR,
2 PSTP,PTEM)
82 80    CALL ADJA (NS,LS,N,ND,RN,PTEM,DIST,LINK,NDEG,M)

```

```

83      CALL COMCON (NS,N,M,RANDOM,PSNO,PSLI,RANN,RANL,
2 LINK,LCOM,NVAR,NV)
84      IF (LCOM.EQ.1) GO TO 90
85      K = 0
86      L = 0
87      DO 184 I = 1,MO
88      DO 181 J = 1,NV
89 181   IF (NMOV(I).EQ.NVAR(J)) GO TO 182
90      GO TO 183
91 182   K = K + 1
92      NMB(K) = NMOV(I)
93      GO TO 184
94 183   L = L + 1
95      NOK(L) = NMOV(I)
96 184   CONTINUE
97      IA = 0
98      IF (K.EQ.0) GO TO 193
99      DO 186 I = 1,NV
100     DO 185 J = 1,K
101 185   IF (NVAR(I).EQ.NMB(J)) GO TO 186
102     IA = IA + 1
103     NAD(IA) = NVAR(I)
104 186   CONTINUE
105     DO 189 I = 1,K
106     J = NMB(I)
107     ISTD(J) = ISTD(J) + 1
108     IF (ISTD(J).GT.MINP) ISTD(J) = MINP
109 189   CALL TEMIN (NS,N,ND,MD,J,DISP(J),ISTD(J),SEP,PSTR,
2 PSTP,PTEM)
110     GO TO 195
111 193   DO 194 I = 1,NV
112 194   NAD(I) = NVAR(I)
113     IA = NV
114 195   IF (IA.EQ.0 .OR. L.EQ.0) GO TO 80
115     DO 187 I = 1,L
116     J = NOK(I)
117     AR(J,1) = J
118 187   AR(J,2) = NDEG(J)
119     CALL SORT (NS,N,AR)
120     IF (IA.GT.L) IA = L
121     DO 190 I = 1,IA
122     DO 191 J = 1,MO
123 191   IF (AR(I,1).EQ.NMOV(J)) GO TO 192
124     GO TO 190
125 192   DO 188 II = 1,ND
126 188   PTEM(J,II) = PSTR(J,II)
127     K = NAD(I)
128     NMOV(J) = K
129     ISTD(K) = 0
130     CALL TEMIN (NS,N,ND,MD,K,DISP(K),ISTD(K),SEP,PSTR,
2 PSTP,PTEM)
131 190   CONTINUE
132     GO TO 80
133 85    CALL ADJA (NS,LS,N,ND,RN,PTEM,DIST,LINK,NDEG,M)

```

```

134 90 IF (FIRST) IC = 0
135 FIRST = .FALSE.
136 DO 93 I = 1,N
137 93 IF (NDEG(I).LT.MNDG) GO TO 92
138 GO TO 99
139 92 DO 94 J = 1,MO
140 94 IF (NMOV(J).EQ.I) GO TO 96
141 II = MO - IC
142 IC = IC + 1
143 IF (II.EQ.0) GO TO 99
144 NH = NMOV(II)
145 NMOV(II) = I
146 DO 95 J = 1,ND
147 95 PTEM(NH,J) = PSTR(NH,J)
148 CALL TEMIN (NS,N,ND,MD,I,DISP(I),ISTP(I),SEP,PSTR,
2 PSTP,PTEM)
149 GO TO 85
150 96 ISTP(I) = ISTP(I) + 1
151 IF (ISTP(I).GT.INT(2.*MD-1)) GO TO 132
152 IONE = 1
153 ONE = 1.
154 CALL TEMIN (NS,N,ND,ONE,I,DISP(I),IONE,SEP,PTEM,
2 PSTR,PTEM)
155 GO TO 85
156 99 DO 110 I = 1,MO
157 J = NMOV(I)
158 IF (J.EQ.0) GO TO 120
159 DO 100 K = 1,ND
160 100 PSTR(J,K) = PTEM(J,K)
161 DISP(J) = DISTAN (NS,J,J,ND,PSTR,PSTP)
162 NFLG(J) = ASTAR
163 110 NMOV(I) = 0
164 120 MO = 0
165 DO 130 I = 1,N
166 130 IF (DISP(I).GE.1.E-4) MO = MO + 1
167 IF (MO.GE.NT) MO = NT
168 IF (MO.EQ.0) FINAL = .TRUE.
169 GO TO 45
170 132 J = 1
171 CALL TEMIN (NS,N,ND,MD,I,DISP(I),J,SEP,PSTR,PSTP,
2 PTEM)
172 DO 135 K = 1,N
173 IF (I.EQ.K) GO TO 135
174 AR(K,1) = K
175 AR(K,2) = DISTAN (NS,I,K,ND,PTEM,PTEM)
176 135 CONTINUE
177 CALL SORT (NS,N,AR)
178 M1 = N - MNDG
179 K = AR(M1,1)
180 DOL = AR(M1,2) - MD + 0.05
181 DO 138 L = 1,ND
182 138 PSTR(I,L) = PTEM(K,L)
183 J = 0

```

```

184      CALL TEMIN (NS,N,ND,DOL,I,AR(M1,2),J,SEP,PTEM,
2 PSTR,PTEM)
185      GO TO 85
186 140   WRITE (6,510) ICASE
187      WRITE (6,501)
188      GO TO 200
189 150   WRITE (6,511)
190      GO TO 200
191 160   WRITE (6,512)
192      GO TO 200
193 170   WRITE (6,513)
      C
      C *****READ FORMATS*****
194 401   FORMAT (40I2)
195 402   FORMAT (2(6X,F7.2),10X,I4)
196 403   FORMAT (2I7/(10X,2F15.6))
      C
      C *****WRITE FORMATS*****
197 501   FORMAT ('1'//15X,'INITIAL AND FINAL POSITIONS OF',
2 'STATIONS'//24X,'INITIAL',11X,'FINAL')
198 502   FORMAT (14X,I5,2F7.1,3X,2F7.1)
199 503   FORMAT ('1'//15X,'NODE      X      Y      DEG      DISTANCE'//)
200 504   FORMAT (16X,I2,A2,2F5.1,I6,F11.3)
201 507   FORMAT (///10X,'PROBABILITY THAT NETWORK SURVIVES ',
2 'RANDOM ATTACK:',F6.3)
202 508   FORMAT (/15X,'TIME INTERVAL',I4)
203 510   FORMAT (//10X,'TERMINAL POSITIONS REACHED IN TIME ',
2 'INTERVAL',I5)
204 511   FORMAT (//15X,'DESIRED TERMINAL POSITIONS ARE NOT ',
2 'CONNECTED USING GIVEN RANGE RESTRICTION')
205 512   FORMAT (//15X,'DESIRED INITIAL POSITIONS ARE NOT ',
2 'CONNECTED USING GIVEN RANGE RESTRICTION')
206 513   FORMAT (//10X,'MAXIMUM NUMBER OF TIME INTERVALS ',
2 'EXCEEDED WITHOUT REACHING DESIRED TERMINAL POSI',
3 'TIONS')
207 515   FORMAT (///18X,'MINIMUM NODE RELIABILITY:',F6.1,
2 ' %'/18X,'MINIMUM LINK RELIABILITY:',F6.1,' %'/24X,
3 'MAXIMUM LINK RANGE:',F7.2/21X,'TOTAL NUMBER OF ',
4 'NODES:',I4/20X,'NUMBER OF NODES MOVING'/25X,'PER ',
5 'TIME INTERVAL:',I4/17X,'MAXIMUM NODE DISPLACEMENT'/
6 25X,'PER TIME INTERVAL:',F7.2/18X,'NUMBER OF RAN',
7 'DOM ATTACKS'/25X,'PER TIME INTERVAL:',I4)
208 516   FORMAT (//19X,'* NODE MOVED DURING CURRENT TIME ',
2 'INTERVAL')
209 517   FORMAT (/13X,'MINIMUM DEGREE (CONNECTIVITY)'/20X,
2 'ALLOWED ON MOVED NODES:',I4)
210 521   FORMAT (//15X,'AVERAGE NODE RELIABILITY:',F8.3/15X,
2 'AVERAGE LINK RELIABILITY:',F8.3)
      C
211 200   STOP
212      END

```

```

213      FUNCTION DISTAN (NSIZE,I,J,ND,PBEG,PEND)
      C
      C      THIS ROUTINE CALCULATES AND RETURNS THE DISTANCE
      C      BETWEEN POINT PBEG(I,K) AND PEND(J,K) FOR
      C      K=1,...,ND.
      C
214      DIMENSION PBEG(NSIZE,ND),PEND(NSIZE,ND)
215      DIS = 0.
216      DO 10 K = 1,ND
217 10    DIS = DIS + (PEND(J,K) - PBEG(I,K))**2
218      DISTAN = SQRT(DIS)
219      RETURN
220      END

```

```

221      SUBROUTINE ADJA (NS,M,N,ND,RN,POSI,DIST,LINK,IDEG,K)
      C
      C      THIS ROUTINE DETERMINES WHICH NODES ARE ADJACENT.
      C
      C      RETURNED VALUES:
      C          LINK(2,K):  TERMINAL NODES OF THE LINKS.
      C          IDEG(N)  :  DEGREES OF THE NODES.
      C          K       :  ACTUAL NUMBER OF LINKS IN NETWORK.
      C
222      DIMENSION DIST(NS,N),POSI(NS,ND),LINK(2,M),IDEG(N)
223      K = 0
224      DO 10 I = 1,N
225      10  IDEG(I) = 0
226      NM1 = N - 1
227      DO 20 I = 1,NM1
228      IP1 = I + 1
229      DO 20 J = IP1,N
230      DIST(I,J) = DISTAN (NS,I,J,ND,POSI,POSI)
231      IF (DIST(I,J).GT.RN) GO TO 20
232      K = K + 1
233      LINK(1,K) = I
234      LINK(2,K) = J
235      IDEG(I) = IDEG(I) + 1
236      IDEG(J) = IDEG(J) + 1
237      20  CONTINUE
238      RETURN
239      END

```

```

240      SUBROUTINE COMPON (NM,N,M,RANDOM,PSN,PSL,RAN,RAL,
2 LINK,LCOM,NV,K)
C
C      THIS ROUTINE DETERMINES THE NUMBER OF COMPONENTS IN
C      THE NETWORK BOTH BEFORE AND AFTER A RANDOM ATTACK.
C      WHEN RANDOM = .FALSE. THE ROUTINE CHECKS THE CON-
C      NECTIVITY OF A DETERMINISTIC NETWORK. WHEN RANDOM =
C      .TRUE. THE CONNECTIVITY IS CHECKED FOR A PROBABILI-
C      STIC NETWORK WHOSE NODES AND LINKS FAIL RANDOMLY.
C
C      RETURNED VALUES:
C          LCOM : NUMBER OF COMPONENTS IN THE NETWORK.
C          NV(K): NODES NOT IN THE MAIN COMPONENT.
C
241      DIMENSION NSUR(50),NNUM(50),NCOM(50),NNEX(50),
2 NLAS(50),NFIR(50),PSN(N),PSL(M),RAN(N),RAL(M),
3 LINK(2,M),AR(50,2),NV(N)
242      LOGICAL RANDOM
243      DO 10 I = 1,N
244          NCOM(I) = 1
245      10 NNEX(I) = 0
246          IF (RANDOM) GO TO 30
247          DO 20 I = 1,N
248              NNUM(I) = 1
249              NFIR(I) = I
250      20 NLAS(I) = I
251          GO TO 55
252      30 DO 50 I = 1,N
253          IF (RAN(I).LT.PSN(I)) GO TO 40
254          NSUR(I) = 0
255          NNUM(I) = 0
256          NFIR(I) = 0
257          NLAS(I) = 0
258          GO TO 50
259      40 NSUR(I) = 1
260          NNUM(I) = 1
261          NFIR(I) = I
262          NLAS(I) = I
263      50 CONTINUE
264      55 DO 100 LN = 1,M
265          IO = LINK(1,LN)
266          JO = LINK(2,LN)
267          IF (.NOT.RANDOM) GO TO 60
268          IF (RAL(LN).GE.PSL(LN)) GO TO 100
269          IF (NSUR(IO).EQ.0 .OR. NSUR(JO).EQ.0) GO TO 100
270      60 IF (NCOM(IO).EQ.NCOM(JO)) GO TO 100
271          IF (NNUM(IO).GE.NNUM(JO)) GO TO 70
272          KO = JO
273          JO = IO
274          IO = KO
275      70 I = NCOM(IO)
276          J = NCOM(JO)
277          IJ = NLAS(I)
278          JJ = NFIR(J)

```

```

279      NNEX(IJ) = JJ
280      NLAS(I) = NLAS(J)
281      NFIR(J) = 0
282      NLAS(J) = 0
283      NNUM(IO) = NNUM(IO) + NNUM(JO)
284      II = NFIR(I)
285  80    NNUM(II) = NNUM(IO)
286      II = NNEX(II)
287      IF (II.NE.0) GO TO 80
288  90    NCOM(JJ) = I
289      JJ = NNEX(JJ)
290      IF (JJ.NE.0) GO TO 90
291  100   CONTINUE
292      LCOM = 0
293      DO 110 I = 1,N
294      IF (NFIR(I).EQ.0) GO TO 110
295      LCOM = LCOM + 1
296      AR(LCOM,1) = I
297      AR(LCOM,2) = NNUM(I).
298  110   CONTINUE
299      IF (RANDOM) RETURN
300      IF (LCOM.EQ.1) RETURN
301      CALL SORT (NM,LCOM,AR)
302      J = 0
303      K = 1
304      DO 130 I = 2,LCOM
305      J = J + K
306      K = AR(I,2)
307      K = J + K - 1
308      IJ = AR(I,1)
309      IJ = NFIR(IJ)
310      DO 130 II = J,K
311      NV(II) = IJ
312  130   IJ = NNEX(IJ)
313      RETURN
314      END

```

315 SUBROUTINE SORT (NM,N,AR)

C
C
C
C
C
C
C
C
C

THIS ROUTINE SORTS THE ELEMENTS OF AN ARRAY IN
DESCENDING ORDER.

RETURNED VALUES:

AR(NM,2): THE SECOND ELEMENT IS THE
SORTED IN DESCENDING ORDER. THE
FIRST ELEMENT OF THE NEW IS THE

316 DIMENSION AR(NM,2)
317 LOGICAL FLAG
318 NM1 = N - 1
319 20 FLAG = .FALSE.
320 DO 60 I = 1,NM1
321 J = I + 1
322 IF (AR(I,2).GE.AR(J,2)) GO TO 40
323 FLAG = .TRUE.
324 DO 40 K = 1,2
325 HOLD = AR(I,K)
326 AR(I,K) = AR(J,K)
327 40 AR(J,K) = HOLD
328 60 CONTINUE
329 IF (FLAG) GO TO 20
330 RETURN
331 END

```

332      SUBROUTINE TEMIN (NM,N,ND,DM,I,DS,IG,SEP,PS,PF,TP)
      C
      C      THIS ROUTINE CALCULATES THE TEMPORARY POSITION FOR
      C      THE ITH NODE AND INSURES THAT IT IS NO CLOSER THAN
      C      SEP TO ANY OTHER NODE.
      C
      C      RETURNED VALUES:
      C      TP(I,ND):  TEMPORARY POSITION OF THE ITH NODE.
      C
333      DIMENSION PS(NM,ND),PF(NM,ND),TP(NM,ND),MT(50)
334      IF (DS.LE.DM) DS = DM
335      DO 100 J = 1,ND
336 100    TP(I,J) = PS(I,J)+(PF(I,J)-PS(I,J))*DM/(DS*2**IG)
337      DO 5 K = 1,N
338      5    MT(K) = 0
339 10    DO 90 J = 1,N
340      10    IF (I.EQ.J) GO TO 90
341      D = DISTAN (NM,I,J,ND,TP,TP)
342      IF (D.GE.SEP) GO TO 90
343      IF (D.LT.1.E-5) GO TO 30
344      DO 20 K = 1,ND
345 20    TP(I,K) = TP(I,K)-(TP(J,K)-TP(I,K))*(SEP-D)/D
346      GO TO 50
347 30    DO 40 K = 1,ND
348 40    TP(I,K) = TP(I,K)-(TP(J,K)-TP(I,K))*SEP
349 50    D = DISTAN (NM,I,I,ND,PS,TP)
350      IF (D.LE.DM) GO TO 90
351      IF (MT(J).EQ.6) GO TO 70
352      MT(J) = MT(J) + 1
353      DO 60 K = 1,ND
354 60    TP(I,K) = TP(I,K)-(PS(I,K)-TP(I,K))*(DM-D)/D
355 70    IF (MT(J).EQ.10) RETURN
356      MT(J) = MT(J) + 1
357      DO 80 K = 1,ND
358 80    TP(I,K) = TP(I,K)-(PS(I,K)-TP(I,K))*DM-D-SEP)/D
359      GO TO 10
360 90    CONTINUE
361      RETURN
362      END

```